

## **PATENT APPLICATION**

### **Pipeline Scalable Architecture for High Density and High Speed Content Addressable Memory (CAM)**

Inventor:                   Xiaohua Huang  
                                  12897 Regan Lane,  
                                  Saratoga CA95070

Entity:                    small

Xiaohua Huang

Tel: 408 396 8285

Fax: 408 868 9150

## **Pipeline Scalable Architecture for High Density and High Speed Content Addressable Memory (CAM)**

[1] This application claims the benefit of provisional U.S. patent Application Serial No. 60/414,030 entitled "Pipeline Scalable Architecture for High Density and High Speed Content Addressable Memory (CAM) Design", filed September 26, 2002 which is incorporated herein by reference in its entirety for all purposes.

### **FIELD OF THE INVENTION**

[2] The present invention is related to content addressable memory. In particular, The invention is related to the Pipeline scalable architecture with hierarchy address decoding and priority encoding.

### **BACKGROUND OF THE INVENTION**

[3] Brief Description of CAM

Basically, CAM is a memory like SRAM or DRAM, which stores  $M$  words and each word is  $N$  bits wide, so the total capacity of the memory will be  $M \times N$  bits.

Besides that CAM can perform simultaneous comparison for a  $N$  bit input with all the  $M$  words stored in the memory. If one of the  $M$  words is equal to the input content on every bit, we say they are matching, and the device will indicate a hit and also give the address in which the matched word is stored.

If none of the  $M$  words is equal to the input content, the device will indicate a miss. If more than one word are equal to the input content, usually the device will pick up the address with high priority and indicate a multi-hit.

Up to now, we got a picture that a CAM needs three functions,

- 1) memory function, which is just like a regular SRAM, with read and write ability,
- 2) comparison or search which can perform simultaneous comparison between an input content and all the M word stored in the memory.
- 3) priority encoding, which picks up the address that has highest priority if more than one match or hit happens.

Fig.1 is the functional block diagram of CAM.

For two Meg bit CAM, if each word is 128 bit wide, there will be 16 K word.

If we put every thing in one block as shown in fig. 1. The device will run very slow.

The reason for this is as follows:

- a) For read and write. The address decoding needs one cycle and cannot be further pipelined. Because 16 K word addresses the address line has huge loading, also the address line itself is also very long. Both wire resistance and loading capacitance, so the RC delay is huge.
- b) Both read and write bit line will be very long and 16 K device loading, and for the same RC delay reason, will be very slow.
- c) The match data bit line will be long and also has 16 K device loading and RC delay will be large.
- d) The priority encoding will be slow. With 16 K input, that is a huge series logic process. It will take a long time. Assume we use hierarchy multilevel encoding, we still need to finish all the encoding within one cycle.

The cycle time will be long.

For the reasons discussed above, we came out with the invention, which will be

described in this filing in the following.

## **SUMMARY OF THE INVENTION**

[4] The Inventions divide the entire CAM block into many identical small sub-block and then symmetrically place them. Divide the address, data bus routing, address decoding, content matching, priority encoding, hit result reading out in different cycle.

In this kind pipe line way, each cycle time can be short, and the throughput of CAM matching can be increased. In this design the power can be reduced. The sub block searching can be achieved. The foregoing, together with other aspects of this invention, will become more apparent when referring to the following specification, claims, and accompanying drawings.

## **BRIEF DESCRIPTION OF THE DRAWINGS**

- [5]
1. Fig.1 The conventional CAM functional diagram
  2. Fig.2 The Hierarchy scalable pipe line CAM architecture

## **DESCRIPTION OF THE SPECIFIC EMBODIMENTS**

[6] The Floor Plan

Here we take 2 Meg bit SRAM based CAM as example (for ternary, the SRAM is four Meg bit, but the principle is the same). We assume the width of each word is  $128=2^7$  bit, so total  $2 \times 2^{20} / 2^7 = 16 \times 2^{10} = 2^{14}$  word. We need 14 bit address to identify each word location. We divide the entire memory into  $256=2^8$  small sub blocks, so each sub block has  $2^{14} / 2^8 = 2^6 = 64$  word. The floor plan is shown in fig. 2. We further divide the 256 sub block into four quadruple. Each quadruple has  $8 \times 8 = 64$  sub-block as shown in fig. 2. The four quadruples are arranged symmetrically. Each sub-block has only 64 word, it is a small SRAM or small CAM, and it can run fast. For 4 Meg bit or 8 Meg bit, the small

sub-blocks become 128 word, or 256 word. Still, can run quite fast.

#### [7] The Bus Routing

As shown in Fig. 2, all the addresses and data and control signals are input from the PAD which are located near the boundary of the chip. First step in routing the signal from the pad at each side (four sides) to the mid point of that side shown as route, (1) in fig. 2, and then buffered. Second step routing each group signal of the four sides to the center of the chip shown in fig 2 is (2) only one route is drawn, the third route is from the center to the mid point of each side of the chip.

Marked as (3) only one was shown in fig. 2.

The fourth step, the signal in route (3) are decoded ( on SRAM read and write case), and then sent to one of the eight columns in one of the quadruples, as route (4).

Signal route (4) further decoded into one of the eight sub-blocks in the column.

The signal of route (4) could be buffered at the starting point of route (4).

For the CAM searching function, no decoding are required and the route (3) signal will be buffered into all the eight columns in each quadruple and then written into each sub-block in each column.

#### [8] Multi-level Decoding

For SRAM operation, read and write, first of all we need to find the address.

in the 2 Meg bit example as we discussed above, total 14 bits address, each sub-block has 6 bit address, and 8 bit are for 256 blocks. We name them as A7, A6, A5, A4, A3, A2, A1, A0. For a given particular address, it is a unique combination of all 14 bit address and corresponding a particular word,

here we are concentrated in finding the sub-block in which that particular word is located. First level decoding is in the center of the chip between route (2) and route (3) then decide the address is in the left or right side. It is decided by Bit A7, we arranged it as if  $A7 = 1$ , the address is on the right side. And if the  $A7 = 0$ , the address is on the left side. In route (3), if  $A6 = 1$  the address is in the upper side (quadruple I or II), if  $A6 = 0$ , the address in the lower side (quadruple III, or IV). {A5, A4, A3} together decide which one column out of 8 column. Based on common 3 to 8 decoding. In route (4), {A2, A1, A0} together decide which block out of 8 blocks are in that column.

After decoding, in each sub-block, it is just like the small block SRAM, CAM design perform read and write, and search for comparisons.

#### [9] Multi-level Muxing

After the block decoding, the data can be written into that block. For read case, the data read out will take the read data bus in route (4) while the other block without reading will not take the bus. Then this column in route (4) will take the read data bus in route (3). Then in route (2) and (1), route (1) is single bus no further muxing. The route (3) and route (4) read data bus will achieve the function described above easily through self-resetting dynamic circuit design.

#### [10] Multi-level Priority encoding.

For CAM searching operation, the input content will be written into each block and compared with each word in every block. So the input data bus through route (1), (2), (3), (4), do not perform any decoding. After compared inside each block,

the matching result should be read out, also needs to perform priority encoding among 256 sub-blocks if multi-hit in one sub-block or hit happens in different blocks.

First step priority encoding (8 to 1) in route (4). The block has highest priority hit will catch the hit result bus, and then the hit address will take the bus. Second step priority among 8 column (8 to 1) in route (3), the highest priority hit column will take the bus and then the hit address in that column will take the hit result bus in route (3).

Step 3, from route (3), to route (2), it is 2 to 1 priority encoding, then in route (2) and route (1) no further encoding.

#### [11] Pipeline Design

Based on the description from section [6] to section [10], we can implement pipeline design in the following way:

make the path from route (1) to route (4) for address decoding, or CAM data input as the first cycle. The sub-block access (read, write, or CAM searching) as the second cycle. And the read data muxing and CAM hit-result priority encoding from route (4) to route (1) as the third cycle. So the SRAM read, SRAM write and CAM search functions can be achieved with three cycle pipe line operation. If high clock rate are required, we can further divide it into more cycles. Say: address decoding, or CAM data input divided into two cycles. Route (1) and route (2) as first cycle. Route (3) and route (4) as second cycle. Block access can also be further divided into two cycles. Read data out and CAM search result address out and priority encoding can be further divided into two cycles. Route (4) and route (3) as one cycle, route (2) and route (1) as another cycle. Total operation will be six cycles.

## [12] Scalable Design

The design described from section[6] to[10], is a scalable design. First , the word number of each sub-block can be changed and will not affect the logic and bus design among each sub-blocks. Second, Without change each sub-block, we can use each sub-block as a basic unit to build one quadruple, or two quadruples, or even partial the column.

If we want to have a larger design, we can re-arrange the floor plan and logic partition among each sub-block and increase the block number. In this way, in Silicon process, a few masks can be saved and cost will be reduced. In the design, the sub-block and bus logic can be re-used for different products, man power can be saved.

## [14] In summary

The design described above are for SRAM based content addressable memory (CAM). It is also applied for ternary CAM(TCAM), or DRAM or psudo-SRAM based CAM. All the inventions or points described from section [4] to [12] will be claimed in the following section.

## **WHAT IS CLAIMED IS:**

1. The LARGE CAM or TCAM are divided into  $2^N$  same size small sub-block and each small sub-block has its own address decoding and priority encoding function.
2. The small sub-block are placed symmetrically around the center of the CAM unit..
3. The small sub-block are equally placed in the each quadruples .